2.1 DESCRIPTION OF COMMANDS

In this chapter each command is described in alphabetical order. These commands are normally entered from the keyboard but they may be incorporated into user programs as described in Section 2.2.2. Each description begins with the following information:

Command line from keyboard	eg A xxxx yyyy
Command in ZEAP-type assemblers	eg SCAL "A
Command in other assemblers	eg RST 18H:DEFB 41H
Command in machine code	eg DF 41

ARITHMETIC COMMAND

```
A xxxx yyyy
SCAL "A
RST 18H:DEFB 41H
DF 41
```

This performs simple arithmetic in hexadecimal, then returns to NAS-SYS. Three results are displayed, viz:

SSSS DDDD JJ where SSSS = yyyy + xxxx
DDDD = yyyy - xxxx
JJ = the displacement required in a Jump Relative instruction which starts at location xxxx and is to jump to yyyy. If the displacement required is too large, ?? is displayed.

e.g. A23 35
0058 0012 10

BREAKPOINT COMMAND

```
B xxxx
SCAL "B
RST 18H:DEFB 42H
DF 42
```

Argument xxxx is stored in workspace location BRKADR (0C23/4), and then the command returns to NAS-SYS.

Following this command, when the user program reaches address xxxx the program execution is stopped and the current contents of the cpu registers displayed. The user program may be restarted from the breakpoint address by giving the command 'E', ie. the Execute command without an argument.

See also the S command.

An Execute command following a B command causes the byte in location xxxx to be replaced by E7, the RST 20 instruction. The original contents of location xxxx is stored in location BRKVAL (0C25). If the breakpoint is reached during execution of the user

program, a jump is made to location $NMI (0C7D/E/F). These locations are initialised by NAS-SYS 3 to contain the instruction JP TRAP. The TRAP routine restores the byte at the breakpoint, saves the cpu registers in the register save area of the NAS-SYS workspace and displays the contents of the registers in the same format as the single step command (refer to S command). Breakpoints may be set only at the first byte of an instruction since the RST 20 can only replace an existing operation code.

COPY COMMAND

```
C xxxx yyyy zzzz
SCAL "C
RST 18H:DEFB 43H
DF 43
```

The Copy command copies memory locations in the block xxxx to xxxx+zzzz-1 to the block yyyy to yyyy+zzzz-1. The transfer begins by writing location xxxx to location yyyy and increments through the block. Note that it is possible to lose data if the two areas overlap.(The intelligent copy command,I,avoids this possibility.)

To fill memory locations xxxx to xxxx+zzzz-1 with the same byte, put the required byte into location xxxx using the M command, then use the C command C xxxx xxxx+1 zzzz. eg. to fill 0E00 to 0EFF with 33, deposit 33 into location 0E00 using M0E00 then C E00 E01 FF.

JUMP TO D000H (NAS-SYS 3 only)

```
D
```

The D command simply starts executing the code beginning at location D000. This will normally be the start of a proprietary program in rom such as ZEAP.

EXECUTE COMMAND

```
E xxxx
SCAL "E
RST 18H:DEFB 45H
DF 45
```

Execute the code beginning at location xxxx. If xxxx is omitted the address used is that stored in the register save area of the NAS-SYS workspace at location RPC (0C69/A).

Arguments may be entered into a program using this command: see section 4.4, Program 4.

Note that the command does **not** simply load the program counter with xxxx and allow program execution from there. The first instruction of a user program will generate an NMI as though that instruction were being single-stepped. However, the contents of the cpu registers are not displayed as they would be using the S, single-step command because workspace location CONFLG (0C26) is set to a non-zero value by the E command, so signalling to NAS-SYS that the E command was called rather than the S command. User program instructions other than the first are executed normally until a breakpoint (if set) is reached.

Since an interrupt will move the cpu past a HALT instruction and the first instruction of the user program when Executed always generates an interrupt, it follows that if the first instruction is HALT, the NASCOM will not in fact halt. To get round this, program a NOP as the first instruction.

F

This command letter is not used in NAS-SYS; a call is made to the routine which displays Error.

GENERATE COMMAND

G xxxx yyyy zzzz
SCAL "G
RST 18H:DEFB 47H
DF 47

Generate writes a header comprising some commands to NAS-SYS followed by program bytes from memory locations xxxx to yyyy-1 on to a cassette tape. When the tape is subsequently read, memory locations from xxxx to yyyy-1 are loaded from the tape and program execution automatically started at location zzzz.

The output to the tape is:

(cr)	Newline
E0 (cr)	Restart NAS-SYS
R (cr)	Read a tape
data	Program code as in W command
E zzzz (cr)	Execute from location zzzz

A program stored on tape using this command can be loaded and run without any commands from the keyboard since the necessary commands are input from the cassette itself.

HALF DUPLEX COMMAND

H
SCAL "H
RST 18H:DEFB 48H
DF 48

The H command causes the Nascom simply to input characters from the devices in the input table and output them to the devices in

the output table.
With the normal tables in operation, the Nascom simply inputs characters from either the keyboard or the serial input port and outputs them to the video display. By using the command X 30 before the H command, input is accepted both from the keyboard and from an external serial device (this latter using device handler XKBD, chapter 3) and outputs to an external serial device (using device handler XOUT, chapter 3) as well as to the video display. Half duplex communication may thus be established with another serial input-output device.

Since the routine executes INLIN (chapter 3) repeatedly, the only way to exit from this command is to reset the Nascom.

INTELLIGENT COPY COMMAND

```
I xxxx yyyy zzzz
SCAL "I
RST 18H:DEFB 49H
DF 49
```

The I command copies memory locations in the block xxxx to xxxx+zzzz-1 to the block yyyy to yyyy+zzzz-1. Unlike the Copy command, if the two blocks overlap, data is not lost.

JUMP TO FFFA

```
J
SCAL "J
RST 18H:DEFB 4AH
DF 4A
```

The J command simply starts executing the code at location FFFA. This is the cold start entry point for 8K ROM BASIC. See also the Z command.

KEYBOARD COMMAND

```
K x
SCAL "K
RST 18H:DEFB 4BH
DF 4B
```

The Keyboard command changes the characteristics of the keyboard.

x	Effect
0	Shift key produces lower case characters.
1	Shift key produces upper case characters.
4	Effect of Graphics key is reversed: keys produce graphics characters unless Graphics key is pressed when alphanumeric characters are produced.
5	Combines the effect of 1 and 4.

See routine KBD, chapter 3.

LOAD COMMAND
(NAS-SYS 1)

```
L
SCAL "L
RST 18H:DEFB 4C
DF 4C
```

This command is used to read data stored on paper tape using the NAS-SYS 1 Tabulate command. The checksum which is output by the Tabulate command is used to check the integrity of the input data. Where the checksum indicates an error, that line is scrolled up the screen so that those locations may be corrected by the M command.

MODIFY COMMAND

```
M xxxx
SCAL "M
RST 18H:DEFB 4D
DF 4D
```

The memory modify command allows memory locations to be displayed and, if the memory is read/write, allows the contents of the location to be changed.

The display shows xxxx followed by the current contents of location xxxx.

To change the contents of the location, simply type in the new hexadecimal value, followed by Enter. The location contents are changed following the Enter.
Where several consecutive bytes are to be modified, as when entering a program in machine code, the bytes may be typed on one line with a space between them. Do not however type beyond the end of a display line before typing Enter. A leading zero does not need to be typed, eg 0C 04 may be typed as C 4.

To exit from the M command, type a full-stop followed by Enter.

To go back to the previous address, type a colon (:) followed by Enter.

To examine a non-consecutive location, yyyy, type /yyyy followed by Enter.

To enter ASCII codes, as when programming a message, type a comma before each character of the message, eg ,A,*,b loads consecutive locations with 41H, 2AH, 62H which are the corresponding ASCII codes.

Invalid characters are indicated by the display 'Error'.
